



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,636	11/25/2003	Robert M. Zeidman		4253

7590 07/31/2007
Robert M. Zeidman
15565 Swiss Creek Lane
Cupertino, CA 95014-5452

EXAMINER

RUTTEN, JAMES D

ART UNIT	PAPER NUMBER
----------	--------------

2192

MAIL DATE	DELIVERY MODE
-----------	---------------

07/31/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/720,636	Applicant(s) ZEIDMAN, ROBERT M.	
	Examiner J. Derek Rutten	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 16 May 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-22 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-22 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 16 May 2007 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date <u>5-16-07</u> | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to Applicant's submission filed 5/16/07, responding to the 3/30/07 Office action which detailed the rejection of claims 1-22. Claims 1, 3-5, 7, 9, 11-16, 18, 20, and 22 have been amended. Claims 1-22 remain pending in the application and have been fully considered by the examiner.

Response to Arguments

2. The objections to the drawings and the claims are withdrawn in view of the 5/16/07 amendments.
3. The rejections of claim 13 under 35 U.S.C. 112, second paragraph, and claims 1-22 under 35 U.S.C. 101 are withdrawn in view of the 5/16/07 amendments.
4. Applicant's arguments filed 5/16/07 have been fully considered but they are not persuasive.

On pages 13-16 (especially the bottom of page 15), Applicant argues with respect to claims 1 and 12, that prior art of record Arnow and Worthington do not teach "eliminating all lines of functional programming code that consist entirely of programming keywords." However, as indicated in the following rejections of claims 1 and 12 below, Arnow discloses elimination of keywords. See page 13, 2nd column, 2nd paragraph. Any line consisting entirely of keywords would be removed. Therefore, Applicant's argument is not persuasive.

On pages 16 and 17 (especially page 17), Applicant argues with respect to claims 5 and 16, that prior art of record, Arnow, Clough, and Worthington do not teach "eliminating all words from said second array of text strings that are less than a minimum length of characters."

Art Unit: 2192

However, as indicated in the following rejections of claims 5 and 16 below, Clough teaches eliminating words that are less than a minimum length of characters. See top of page 14.

Therefore, Applicant's argument is not persuasive.

On pages 17-19 (especially page 18), Applicant argues with respect to claims 7, 8, 11, 18, 19, and 22, that prior art of record Arnow, Clough, and Worthington, do not teach the claim limitations. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "identifiers that have not been modified from their original form" – see page 18 of Applicant's response) are not recited in the rejected claims. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Applicant further argues that Clough teaches away from the invention since matching is accomplished using an encoded string. However, Clough does not criticize, discredit, or otherwise discourage any solutions for matching substrings. To the contrary, Clough meets the plain language of the claims by finding partial matches between text strings. Therefore, Applicant's arguments are not persuasive.

On pages 19-20, Applicant provides four arguments regarding the rejections of claims 9, 10, 20, and 21:

(1) Applicant argues that the claims at issue are not directed to "keywords," and therefore the Wegmann reference does not apply. However, Arnow discloses matching keywords, and Wegmann is used to show that certain keywords are found at the beginning of a line. The combination of references teaches that words at the beginning of a line are matched using keywords. Therefore, Applicant's arguments are not persuasive.

(2) Applicant argues that Wegmann does not teach that all keywords start at the beginning of a line. However, Wegmann does teach that all keywords whose templates can be inserted start at the beginning of a line. The combination of Arnow and Wegmann teaches finding matches at the beginning of a line. Therefore, Applicant's arguments are not persuasive.

(3) Applicant argues that Wegmann uses a specific programming language that one of ordinary skill would not generalize to other languages. However, the plain language of the claim simply calls for "finding sequences where the first word of each line in said first array matches the first word of each line in said second array." The combination of Arnow and Wegmann teach finding such a match. Therefore, Applicant's arguments are not persuasive.

(4) Applicant argues that GNU Emacs as used by Wegmann is simply a text editor that "has no particular use in finding plagiarism or comparing sequences of lines of code." Applicant references the online encyclopedia Wikipedia <<http://www.wikipedia.org>> to support the argument. However, the 7/18/07 revision of the Wikipedia entry for Emacs included a description of extension libraries including "Ediff" and "Emerge" which are both used in Emacs to compare files (see the "Features" section). Thus, it appears that Emacs is in fact used for comparing sequences of lines of code. Therefore, Applicant's arguments are not persuasive.

Information Disclosure Statement

5. The information disclosure statement (IDS) submitted on 5/16/07 was filed after the mailing date of the first Office action on 3/30/07. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 3, 4, 14, and 15 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 3 has been amended to be dependent upon claim 1. In lines 3-4, claim 3 recites "eliminating functional programming lines from said first source code file, leaving comment lines." Similar limitations are present in lines 5-6. However, claim 1 contains limitations removing comments from the source code files. Since comments were removed in claim 1, it is not clear how any comment lines would remain in claim 3. Further, claim 1 recites "putting each remaining line of functional programming code of the first file into a first array of text strings." This "putting" can be interpreted as a cut and paste operation where the functional programming code no longer remains in the first file. In this case, there would be no functional programming lines left to eliminate as recited in claim 3. Claim 14 contains limitations similar to claim 3 and is rejected for the same reasons. Claims 4 and 15 are rejected as being dependent upon a rejected

Art Unit: 2192

base claim. For the purpose of further examination, claims 3 and 14 will be interpreted as having source code files with remaining comments and functional programming code.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claims 1, 3, 12, and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over “:-
)When you grade that: using e-mail and the network in programming courses” by Arnow
(hereinafter “Arnow”) in view of “Strings” by Worthington (hereinafter “Worthington”).

In regard to claim 1, Arnow discloses:

A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, (See page 13, 2nd column, 2nd paragraph, e.g. “plagiarism detector... carries out a number of transformations”) the method comprising eliminating programming comments from the first source code file; eliminating programming comments from the second source code file; See page 13, 2nd column, 2nd paragraph. This provides a description of separating program code into two files, one of

which containing only programming code and no comments. This is done for “each student’s code.” That is, a first and second source code file.

substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said first file; substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said second file; See page 13, 2nd column, 2nd paragraph, e.g. “spaces and tab sequences are reduced to single spaces.”

eliminating all lines of functional programming code in said first file that consist entirely of programming keywords; eliminating all lines of functional programming code in said second file that consist entirely of programming keywords; See page 13, 2nd column, 2nd paragraph, e.g. “one containing only the identifiers, the other only the keywords and operators.” Note that removal of keywords inherently removes lines that consist entirely of keywords.

...

finding all matches between text strings in said first array with text strings in said second array; See page 13, 2nd column, 2nd paragraph, e.g. “pairwise comparison.”

and displaying a report showing said matches. See page 13, 2nd column, 2nd paragraph, e.g. “diff utility.” Arnov discloses use of the diff utility which displays reports of matches and differences. Note that this inherent quality is described in the prior art of record “Diff, Patch, and Friends” by Johnson (see at least the displayed report shown at the top of page 2).

Arnow does not expressly disclose: *putting each remaining line of functional programming code of the first file into an array of text strings; putting each remaining line of functional programming code of the second file into a second array of text strings;* However, Worthington teaches using arrays to store strings. See top of page 3, e.g. "It is often useful to store strings in arrays." It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Worthington's array of strings to store Arnow's program code in order to utilize the useful nature of arrays as suggested by Worthington.

In regard to claim 3, the above rejection of claim 1 is incorporated. Arnow further discloses:

eliminating functional programming lines from the first source code file, leaving comment lines; b) eliminating functional programming lines from the second source code file, leaving comment lines; See page 13, 2nd column, 2nd paragraph. This provides a description of separating program code into two files, one of which containing only comments and no functional programming. This is done for "each student's code." That is, a first and second source code file.

substituting a single space character for sequences of whitespace characters in each remaining comment line in said first file; d) substituting a single space character for sequences of whitespace characters in each remaining comment line in said second file; See page 13, 2nd column, 2nd paragraph, e.g. "spaces and tab sequences are reduced to single spaces."

...

finding all matches between text strings in said first array with text strings in said second array; See page 13, 2nd column, 2nd paragraph, e.g. “pairwise comparison.”

and displaying a report showing said comment line matches. See page 13, 2nd column, 2nd paragraph, e.g. “diff utility.” Arnow discloses use of the diff utility which displays reports of matches and differences. Note that this inherent quality is described in the prior art of record “Diff, Patch, and Friends” by Johnson (see at least the displayed report shown at the top of page 2).

Arnow does not expressly disclose: *putting each remaining comment line of the first file into an array of text strings; putting each remaining comment line of the second file into a second array of text strings;* However, Worthington teaches using arrays to store strings. See top of page 3, e.g. “It is often useful to store strings in arrays.” It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Worthington’s array of strings to store Arnow’s program code in order to utilize the useful nature of arrays as suggested by Worthington.

In regard to claim 12, Arnow discloses an apparatus comprising a computer. See page 10 column 2 paragraphs 2 and 3, e.g. “Sun IPCs.” All further limitations have been addressed in the above rejection of claim 1.

In regard to claim 14, the above rejection of claim 12 is incorporated. All further limitations have been addressed in the above rejections of claim 3.

10. Claims 2, 4-8, 11, 13, 15-19, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Arnow in view of Worthington as applied to claim 1 above, and further in view of "Plagiarism in natural and programming languages: an overview of current tools and technologies" by Clough (hereinafter "Clough").

In regard to claim 2, the above rejection of claim 1 is incorporated. Arnow and Worthington does not expressly disclose: *where finding all matches ignores the type case of the text*. However, Clough teaches that the YAP system translates upper-case letters to lower case letters. See page 24 under "Preprocess the submitted reports." It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Clough's case translation with Arnow's program code in order to preprocess text for tokenization as suggested by Clough.

In regard to claim 4, the above rejection of claim 3 is incorporated. All further limitations have been addressed in the above rejection of claim 2.

In regard to claim 5, Arnow discloses:

A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, (See page 13,

Art Unit: 2192

2nd column, 2nd paragraph, e.g. “plagiarism detector...carries out a number of transformations”) *the method comprising*

...

eliminating programming language keywords from said array of text strings; ... eliminating programming language keywords from said second array of text strings; See page 13, 2nd column, 2nd paragraph, e.g. “one containing only the identifiers, the other only the keywords and operators.”

finding all matches between text strings in said first array with text strings in said second array. See page 13, 2nd column, 2nd paragraph, e.g. “pairwise comparison.”

...

displaying a report showing said matches. See page 13, 2nd column, 2nd paragraph, e.g. “diff utility.” Arnow discloses use of the diff utility which displays reports of matches and differences. Note that this inherent quality is described in the prior art of record “Diff, Patch, and Friends” by Johnson (see at least the displayed report shown at the top of page 2).

Arnow does not expressly disclose: *extracting all words between whitespace from each line of functional programming code in the first source code file to an array of text strings; ...extracting all words between whitespace from each line of functional programming code in the second source code file to a second array of text strings; eliminating all words from said first array of text strings that are less than a minimum length of characters; eliminating all words from said second array of text strings that are less than a minimum length of characters.*

However, Clough teaches document comparison based on word use. See bottom of page 11 “show the number of words in common.” Clough also teaches eliminating words that are less than a minimum length of characters. See top of page 14, e.g. “A first lexicon-generating parse eliminates all numbers, words consisting of one or two letters...” It would have been obvious to one of ordinary skill at the time the invention was made, to use Clough’s teaching of word matching with Arnow’s keyword elimination in order to identify documents that contain similar passages, as suggested by Clough (see bottom of page 11). Further, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use Clough’s elimination of words with a minimum character length with Arnow’s elimination in order to provide a useful parse as suggested by Clough (see top of page 14).

Also, Worthington teaches using arrays to store strings. See top of page 3, e.g. “It is often useful to store strings in arrays.” It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Worthington’s array of strings to store Arnow’s program code in order to utilize the useful nature of arrays as suggested by Worthington.

In regard to claim 6, the above rejection of claim 5 is incorporated. All further limitations have been addressed in the above rejection of claim 2.

In regard to claim 7, Arnow and Worthington do not expressly disclose: *e) finding all partial matches between text strings in said first array with text strings in said second*

array, where a partial match is where one string can be found in its entirety in as a second string but the strings are not identical. However, Clough teaches matching substrings. See page 22, bulleted list in section 4.1.3, e.g. “q as a substring of S.” It would have been obvious to one of ordinary skill at the time the invention was made, to use Clough’s teaching of substring matching with Arnow’s plagiarism detector in order to detect systemic changes to variable names as suggested by Clough (see 1st paragraph in section 4.1.3). All further limitations have been addressed in the above rejection of claim 5.

In regard to claim 8, the above rejection of claim 7 is incorporated. All further limitations have been addressed in the above rejection of claim 2.

In regard to claim 11, Arnow discloses:

A method for comparing two program source code files, (See page 13, 2nd column, 2nd paragraph, e.g. “plagiarism detector...carries out a number of transformations”) *comprising:*

extracting from each program source code file a first set of code elements and a second set of code elements; See page 13, 2nd column, 2nd paragraph. This provides a description of separating program code into two files, one of which containing only programming code and no comments. This is done for “each student’s code.” That is, a first and second source code file.

computing a first metric derived from comparing the first set of code elements for the first program source code file to the first set of code elements for the second program source code file; See page 13, 2nd column, 2nd paragraph, e.g., “smaller numbers suggesting plagiarism”

computing a second metric derived from comparing the second set of code elements for the first program source code file to the second set of code elements for the second program source code file; See page 13, 2nd column, 2nd paragraph. Arnow provides for the comparison at least of comments and programming code. As such, at least two metrics are derived.

combining the first metric and the second metric to derive a combined metric, See page 13, 2nd column, 2nd paragraph, e.g. “smaller numbers suggesting plagiarism.” This suggests that for each set of files, Arnow computes a metric and combines them to arrive at a “number” which provides an indication of plagiarism.

wherein the first and second sets of code elements are selected from the group consisting of selected source lines See page 13, 2nd column, 2nd paragraph, e.g. “identifiers”, selected comment lines See page 13, 2nd column, 2nd paragraph, e.g. “comments,” and selected code sequences See page 13, 2nd column, 2nd paragraph, e.g. “keywords”.

displaying a report showing said combined metric. See page 13, 2nd column, 2nd paragraph, e.g. “diff utility.” Arnow discloses use of the diff utility which displays reports of matches and differences. Note that this inherent quality is described in the prior art of record “Diff, Patch, and Friends” by Johnson (see at least the displayed report shown at the top of page 2).

Arnow does not expressly disclose *wherein the first and second sets of code elements are selected from the group consisting of complete words, selected partial words*. However, Clough teaches searching for complete and partial words. See bottom of page 11 “show the number of words in common.” It would have been obvious to one of ordinary skill at the time the invention was made, to use Clough’s teaching of word matching with Arnow’s keyword elimination in order to identify documents that contain similar passages, as suggested by Clough (see bottom of page 11). Also see page 22, bulleted list in section 4.1.3, e.g. “q as a substring of S.” It would have been obvious to one of ordinary skill at the time the invention was made, to use Clough’s teaching of substring matching with Arnow’s plagiarism detector in order to detect systemic changes to variable names as suggested by Clough (see 1st paragraph in section 4.1.3).

In regard to claims 13 and 15, the above rejections of claims 12 and 14 are respectively incorporated. All further limitations have been addressed in the above rejection of claims 2 and 4, respectively.

In regard to claim 16, all limitations have been addressed in the above rejections of claims 5 and 12.

In regard to claim 17, the above rejection of claim 16 is incorporated. All further limitations have been addressed in the above rejection of claim 2.

In regard to claim 18, all limitations have been addressed in the above rejections of claims 7 and 12.

In regard to claim 19, the above rejection of claim 18 is incorporated. All further limitations have been addressed in the above rejection of claim 2.

In regard to claim 22, Arnow discloses an apparatus comprising a computer. See page 10 column 2 paragraphs 2 and 3, e.g. "Sun IPCs." All further limitations have been addressed in the above rejection of claim 11.

11. Claims 9 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Arnow in view of Worthington as applied to claim 1 above, and further in view of "TLA+ Mode: Editing Features" by Wegmann (hereinafter "Wegmann").

In regard to claim 9, Arnow discloses using keywords to find matching code. See page 13, 2nd column, 2nd paragraph, e.g. "the other only keywords and operators." Arnow and Worthington do not expressly disclose: *g) finding sequences where the first word of each line in said first array matches the first word of each line in said second array.*

However, Wegmann teaches that keywords are found at the beginning of a line. See page 4, 3rd paragraph under "Formatting and Indenting," e.g. "The keywords whose templates can be inserted all start at the very beginning of a line." It would have been obvious to one of ordinary skill at the time the invention was made, to use Wegmann's teaching of

Art Unit: 2192

the location of keywords with Arnow's keyword matching in order to easily find the likely location of keywords as suggested by Wegmann. All further limitations have been addressed in the above rejection of claim 1.

In regard to claim 20, all limitations have been addressed in the above rejections of claims 9 and 12.

12. Claims 10 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Arnow in view of Worthington in view Wegmann as applied to claim 9 above, and further in view of Clough.

In regard to claim 10, the above rejection of claim 9 is incorporated. Arnow, Worthington, and Wegmann do not expressly disclose: *where finding sequences where the first word of each line in said first array matches the first word of each line in said second array ignores the type case of the text*. However, Clough teaches that the YAP system translates upper-case letters to lower case letters. See page 24 under "Preprocess the submitted reports." It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Clough's case translation with Arnow's program code in order to preprocess text for tokenization as suggested by Clough.

In regard to claim 21, the above rejection of claim 20 is incorporated. All further limitations have been addressed in the above rejection of claim 10.

Conclusion

13. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a).

Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (571)272-3703. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications

Art Unit: 2192

may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/jdr/



TUAN DAM
SUPERVISORY PATENT EXAMINER